N0VEL TECHNIQUE FOR SIGNAL CLASSIFICATION BASED ON NEURAL NETWORK IN VLSI

Dhananjay S Dabhade
Asst. Professor
Department of Electronics & Telecommunication Engineering
Dhamangaon Educational Society's College of Engg. and Tech., Dhamangaon Rly(MS)

Mangesh S Dabhade
Department of Computer Science Engineering
Government College of Engineering, Amrawati(MS)

Abstract:

Wireless sensor network is highly data centric. Data communication in wireless sensor network must be efficient one and must consume minimum power. Every sensor node consists of multiple sensors embedded in the same node. Thus every sensor node is a source of data. These raw data streams cannot be straightway communicated further to the neighboring node. These sensor data streams are first classified. A group of sensor nodes forms a cluster. Each node transfer data to a cluster head and then cluster head aggregates the data and sends to base station. Hence clustering and classification techniques are important and can give new dimension to the WSN paradigm. Basically, classification system is either supervised or unsupervised, depending on whether they assign new inputs to one of a infinite number of discrete supervised classes or unsupervised categories respectively. ART1 and Fuzzy ART are unsupervised neural network models which are used for classification of sensor data. ART1 model is used for classification of Binary valued data. While Fuzzy ART model can be used for analog data, wherein the input data is fuzzy valued.

Keywords-: Artificial Neural Networks (ANN), Neural Network Architecture (NNA), Multi-layer neural network (MNN).

I. INTRODUCTION

Wireless sensor network (WSN) is an emerging technology that has wide range of potential applications including environment monitoring, smart spaces, medical systems, and robotics exploration. Such networks will consist of large number of distributed nodes that organize themselves into a multihop wireless network[1]. Each sensor node has one or more sensors, embedded processors, and low-power radios, and is normally battery operated. Typically, these nodes coordinate and communicate to perform a common task. These sensor nodes remain largely inactive for long time, but becoming suddenly active when something is detected. An important challenge in the design of these networks is that two key resources - communication bandwidth and energy are significantly more limited than in a tethered network environment. These constraints require innovative design techniques to use the available bandwidth and energy efficiently [2]. The communication consumes the largest part of the energy budget. Hence attempt must be done to implement techniques two save energy on communications. The paper discusses real time classifier using ART1 and Fuzzy ART neural networks model. Real time classifier classifies the sensor readings and then only its class ID needs to be communicated further. This brings a saving of sufficient amount of energy. The paper also deals with the VLSI implementations of ART1 and Fuzzy ART neural models.

II. Building Blocks of Neural Networks

Neural networks are made of basic units (Figure 1) arranged in layers. A unit collects information provided by other units (or by the external world) to which it is connected with weighted connections called synapses. These weights, called synaptic weights multiply (i.e., amplify or attenuate) the input information. A positive weight is considered excitatory, a negative weight inhibitory.

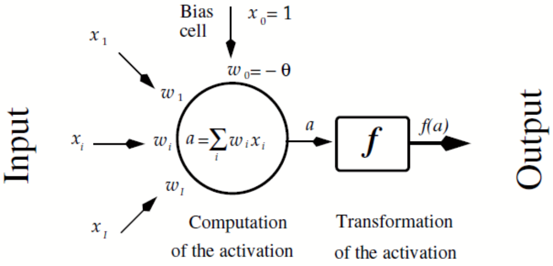


Figure. 1: The basic neural unit

Each of these units is a simplified model of a neuron and transforms its input information into an output response.

Proc. of the Intl. Conf. on Advances in Computer, Electronics and Electrical Engineering
Editor In Chief Dr. R. K. Singh. Copyright © 2012 Universal Association of Computer and Electronics Engineers.
All rights reserved. ISBN: 978-981-07-1847-3

This transformation involves two steps: First, the activation of the neuron is computed as the weighted sum of it inputs, and second this activation is transformed into a response by using a transfer function. Formally, if each input is denoted x_i, and each weight w_i, then the activation is equal to $a = \sum x_i w_i$, and the output denoted o is obtained as o = f(a). Any function whose domain is the real numbers can be used as a transfer function. The most popular ones are the linear function (o α a), the step function (activation values less than a given threshold are set to 0 or to −1 and the other values are set to +1), the logistic function

$$\left[f(x) = \frac{1}{1+\exp\{-x\}} \right] \tag{1}$$

which maps the real numbers into the interval [−1 + 1] and whose derivative, needed for learning, is easily computed *{f'(x) = f(x) [1 − f(x)]}*, and the normal or Gaussian function.

$$o = \left(\sigma\sqrt{2\pi}\right)^{-1} \times \exp\left\{-\tfrac{1}{2}(a/\sigma)^2\right\} \tag{2}$$

Some of these functions can include probabilistic variations; for example, a neuron can transform its activation into the response +1 with a probability of 1/2 when the activation is larger than a given threshold. The architecture (i.e., the pattern of connectivity) of the network, along with the transfer functions used by the neurons and the synaptic weights, completely specify the behavior of the network.

III. ARTIFICIAL NEURAL NETWORKS PARADIGM IN WIRELESS SENSOR NETWORKS

Wireless sensor network is highly data centric. Data communication in WSN must be efficient one and must consume minimum power. Every sensor node consists of multiple sensors embedded in the same node. Thus every sensor node is a source of data. These raw data streams cannot be straightway communicated further to the neighboring node or the base station. This sensor data stream needs to be classified. A group of sensor nodes forms a cluster. Each node transfer data to a cluster head and then cluster head aggregates the data and sends to base station. Hence clustering and classification techniques are important and can give new dimension to the WSN paradigm. Hence efficient data clustering techniques must be used to reduce the data redundancy and in turn reduce overhead on communication. This can be very well accomplished by using some of the Algorithms developed within the artificial neural networks paradigm, which can be easily adapted to WSN. Artificial neural network (ANN) consists of small computing units, called neurons, arranged in different layers and interconnected with each other explained in[8]. Simple mathematical computations are performed at each neuron. There are lots of advantages of using ANN in WSN and will meet the requirements for WSN like - simple parallel-distributed computations, distributed storage and data robustness. Thus it can reduce memory requirement to the minimum. The WSN nodes can be clustered in different groups, thereby reducing the dimension of the network. This lowers communication and memory cost. Carpenter and Grossberg developed ART; it provides a solution for the plasticity and stability dilemma [5]. ART can learn arbitrary input patterns in a stable, fast and self organizing way, thus overcoming the effect of learning in stability that plagues many other competitive networks. ART is not, as is popularly imagined, a neural network architecture. It is a learning theory, that resonance in neural circuits can trigger fast learning. As such it subsumes a large family of current and future neural networks architectures, with many variants. ART1 is the first member, which only deals with binary input pattern [4], although it can be extended to arbitrary input patterns by a variety of coding mechanisms. ART2 and Fuzzy ART extend the applications to analog input patterns [5].

Fuzzy ART (FA) benefits the incorporation of fuzzy set theory and ART [5]. Fuzzy ART maintains similar operations to ART1 and uses the fuzzy set operators, so that it can work for all real data sets. Fuzzy ART exhibits many desirable characteristics such as fast and stable learning and typical pattern detection.

IV. ART1 ALGORITHM SUITABLE FOR VLSI IMPLEMENTATION

The ART1 model is described in Fig. 1 and deled in detail in[10] . Each Ft node xi is connected to all F2 nodes y_i through bottom up connection

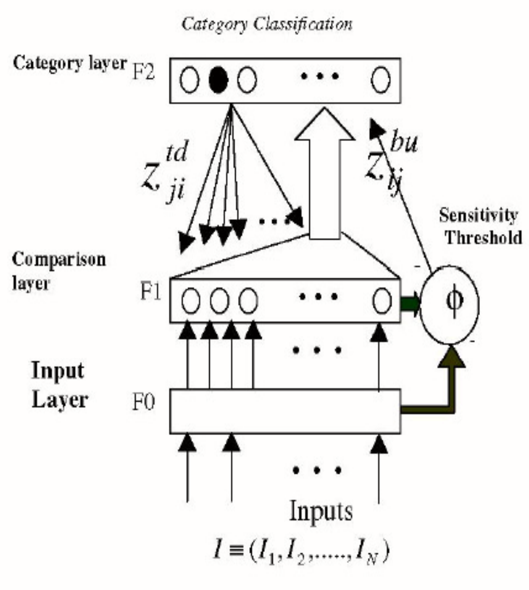


Fig. 2: Architecture of ART1 model

z_{ij}^{bu}

weights , so that the input received by each F2 node y_i is given by

$$T_j = \sum_{i=1} z_{ij}^{bu} I_i \qquad (3)$$

Bottom up weights z_{ij}^{bu} take any real value in the interval [0,K], where

$$K = \frac{L}{L-1+N}, \qquad \text{and} \qquad L > 1$$

Layer F2 acts as Winner-Take-All network, i.e. a competitive layer for the outputs, so that all nodes yi will stay inactive, except the one that receives the largest bottom up input T. Once an F2 winning node arises a top-down template is activated through the top-down weights z_{ji}^{td}. In the fast learning Type-3 model top-down weights z_{ji}^{td} take values ' 0 ' or ' 1 ' . Let us call this top-down template X = (X_1, X_2,......X_N) . The resulting vector X is given by the equation,

$$X_i = I_i \sum_j z_{ji}^{td} y_i \qquad (4)$$

Since only one y_i is active, let us call this winning F2 node Y_j, so that Y_j =o if j≠0 and Y_j =1. In this case we can state

$$X_i = I_i z_{Ji}^{td} \qquad \text{or} \qquad X = I \cap z_J^{td} \qquad (5)$$

where $z_J^{td} \equiv (Z_{1J,}\ Z_{2J,}...,Z_{NN})$ This top-down template will be compared against the original input pattern I according to a predetermined vigilance criteria, tuned by the vigilance parameter 0 < ρ < 1, so that two alternative may occur: (1) If $\rho\ |I| \leq |I \cap z_J^{td}|$ the active category J is accepted and the system weights will be uploaded to incorporate this new knowledge. (2) If $\rho\ |\ I\ | > |\ I \cap z_J^{td}\ |$ the active category J is not valid for the actual value of the vigilance parameter ρ. In this case y_J will be deactivated (reset) making T_j = 0, so that J another y_j node will become active through the Winner-Take-All action of the F2 layer. Here the notation |r| represents the cardinality of vector r, i.e.

$$z_J^{bu}\ |_{new} = \frac{L(I \cap (z_J^{td})_{old})}{L-1+|(I \cap (z_J^{td})_{old}|} \qquad (6)$$

$$z_J^{td}\ |_{new} = I \cap (z_J^{td})_{old}$$

note that only the weights of the connections touching the F2 winning node y_J are updated. This algorithm along with flowchart is described in[12]

V. ART NEURAL NETWORK MODEL

The general structure of the Fuzzy ART neural network is shown in Fig. 2. It consists of two layers of neurons that are fully connected: a 2M-neuron input or comparison layer (F1) and an N neuron output or competitive layer (F2). A weight value Z_{ji} is associated with each connection, where the indices i and j denote the neurons that belong to the layer Ft and F2 respectively. The set of weight Z = {z_{ji}: i = 1,2,...,2M; j = 1,2,...,N} encodes information that defines the categories learned by the network. These can be modified dynamically during network operation. For each neuron j of F2, the vector adaptive weights z_j = ($z_{j1,}$ $z_{j2,.....,}$ z_{j2M}) correspond to the subset of weights ($z_j \subset Z$) connected to neuron j. This vector Z is named prototype vector or template, and it represents the set of characteristics defining the category j. Each prototype vector Z is formed by the characteristics of the input patterns to which category j has previously been assigned through winner-take-all competition.

VI. FUZZY ART ALGORITHM

This algorithm can be described in five execution steps:

A. Weights and parameter initialization

Initially, all the neurons of F2 are uncommitted, and all weight values Z_{ji} are initialized to 1. An F2 neuron becomes committed when it is selected for an input P. Then, the corresponding weights z_{ji} can take values expressed by real number in the interval [0,1]. The Fuzzy ART weight vector Z_{ji} subsumes both the bottom-up and top-down weight vectors of ART1.

B. Input Vector Coding

When a new input vector P (P_1, P_2, ….., P_N) of N elements

$$y_J = 1, \quad \text{if, } T_J = \max_k \{T_k\}, \text{ otherwise } \quad y_J = 0$$

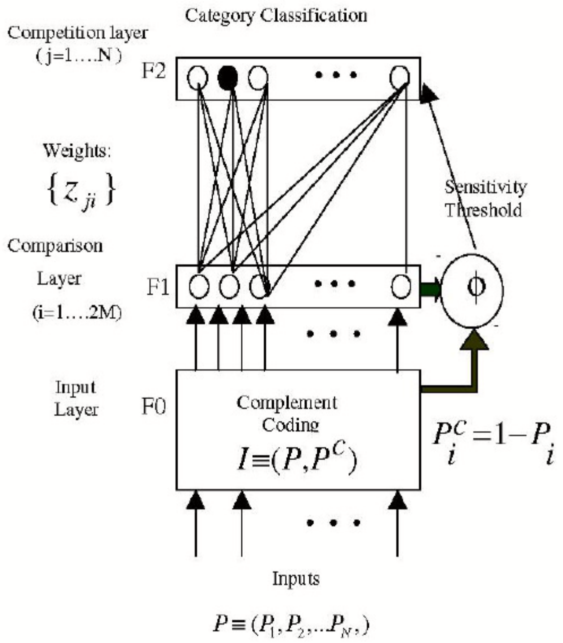


Fig. 3: ART Neural Network Model

(where each element is a real number in interval [0,1] is presented to the network, it undergoes a preliminary coding at layer FO. Complement coding of vector P results in a network input vector I of 2N elements such that: I ≡(P,P^C)= $(P_1, P_2,...P_1 P_1^C, P_2^C,..., P_N^C$),with P_i^c = 1-P_i . This coding is used to prevent a category proliferation when the weights erode.

C. Category Choice

With each presentation of an input I to Fl, the choice function T_j(I) is calculated for each neuron j in F2:

$$T_j = \frac{|I \wedge z_{ji}|}{\alpha + |z_{ji}|} \tag{7}$$

Here the notation |r| represents the cardinality of vector r , i.e. $|r|=\sum_{i=1}^{2N} r_i$, ^ is the fuzzy logic AND operator $I \wedge z_j = (\min(I_1, z_{j1}, \min(I_2, z_{j2})....\min(I_{2N}, I_{j2N}))$ and α is a user defined choice parameter such that α >0. F2 is a winner-taking-all competitive layer, where the winner is the neuron j=J with the greatest value of activation T_j for the input I, T_j = max{ T_j : j = 1,...., N}. If the same T_j value is obtained by two or more neurons, the one with the smallest index j wins. The winning neuron J is retained for steps D and E.

D. Vigilance Test

This step serves to compare the similarity between the prototype vector of the winning neuron z_{ji} and input I, against a user defined vigilance parameter ρ, through the following test:

$$\frac{|I \wedge z_J|}{|I|} \geq \rho \tag{8}$$

where ρ =[0,1]. This comparison is carried out on layer Fl: the winning neuron J transmits its learned expectancy, z_j, to Fl for comparison with I. If the vigilance test (Equ. 7) is passed, then neuron J becomes selected and is allowed to adapt its prototype vector as per step E. Otherwise, neuron J is deactivated for the current input I: T_J is set equal to -1 for the duration of the current input presentation. The algorithm searches through the remaining F2 layer neurons (steps C and D), until some other neuron J passes the vigilance test. If no committed neuron from the F2 layer can pass this test, an uncommitted neuron is selected and undergoes prototype vector update (i.e. the new class is assigned for the input)

E. Prototype vector update

The prototype vector of the winning neuron J is updated according to:

$$z_J^{new} = \beta(I \wedge z_J^{old}) + (1-\beta) z_J^{old} \tag{9}$$

Where β is a user defined learning rate parameter such that β = [0,1]. The algorithm can be set to slow learning, with 0< β <1. For fast learning β =1, the new weight can be updated as - $z_J^{new} = (I \wedge z_J^{old})$. Once this update step is accomplished, the network can process a new input vector from step B.

VII. Features of ART1 and FuzzyART Model Useful For VLSI Implementation

The modified ART1 and FuzzyART algorithm discussed in this paper can be easily implemented in VLSI. The mathematical processing of the algorithm is modified to simplified algebraic operations. Fuzzy ART can be considered as a particular case of ART1, when inputs are binary. T_j which is a distances or choice functions in case of Fuzzy ART can be used in ART1. This further simplifies the ART1 algorithm.

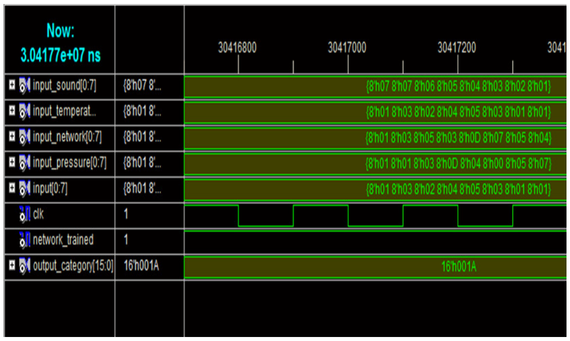
ART1 and FuzzyART algorithm bears many features to be implemented as VLSI unit:

(1) The sensor network needs real time clustering, ART1 and FuzzyART can be used for real time clustering of binary input patterns and analog as well. VLSI circuits can very easily handle binary input patterns.

(2) The inter connection weights of the ART1 and FuzzyART neuron are binary valued and the resolution of the weights is not affected by the size nor the storage capacity of the system. This and the non necessity of analog weights is one of the most hardware attractive feature of the ART1 and FuzzyART algorithm.

(3) ART1 and FuzzyART neuron is self scaling: some neural model need to increase their weight resolution as they scale up and they need to scale up size and interconnectivity with pattern size or storage capacity. Hardware implementation cannot afford such flexibility, may be possible at software platforms. For ART1 and FuzzyART model , as shown in Figure 3.1 the number of neurons N in the bottom layer is the number of pixels of the input pattern, the number of neurons M in the top layer is the maximum number of categories and N x M is the number of synapses (connections). This system scales up linearly with storage capacity (M) and input pixels (N). For a bidirectional associative Memory (BAM) neural model, the size scales quadratically with the storage capacity and the number of pixels.

VII. SIMULATION RESULTS



```
Now:                                       30416800        30417000        30417200        3041
3.04177e+07 ns
input_sound[0:7]       {8'h07 8'...        {8'h07 8'h07 8'h06 8'h05 8'h04 8'h03 8'h02 8'h01}
input_temperat...      {8'h01 8'...        {8'h01 8'h03 8'h02 8'h04 8'h05 8'h03 8'h01 8'h01}
input_network[0:7]     {8'h01 8'...        {8'h01 8'h03 8'h05 8'h03 8'h0D 8'h07 8'h05 8'h04}
input_pressure[0:7]    {8'h01 8'...        {8'h01 8'h01 8'h03 8'h0D 8'h04 8'h00 8'h05 8'h07}
input[0:7]             {8'h01 8'...        {8'h01 8'h03 8'h02 8'h04 8'h05 8'h03 8'h01 8'h01}
clk                    1
network_trained        1
output_category[15:0]  16'h001A            16'h001A
```

Fig 4. Output with category

Explanation:

Figure 4. above shows the simulation result for the complete classifier. Here the inputs for each of the parameters i.e., temperature, weights, network and pressure are multiplied with their respective weights. The sum of the products obtained for each of the parameters is then compared with their respective thresholds. If sum output matches with any of the sensitivity threshold value, then category of input signal is obtained. If there is no match then input signal is a noise.

VIII CONCLUSION

Communication is the largest consumer of energy budget in wireless sensor networks. Clustering sensor nodes in different architectures and then classifying sensor data at each node can save a considerable amount of power. Energy saving can be of the order of number of classes of sensor data at each node. The number of classes should never exceed the value s * p. The popular ART1 and Fuzzy ART algorithms are transformed to suite for VLSI implementation using Model-sim & XILINX 9.2i. The modified ART1 and Fuzzy ART are tested on MATLAB with different patterns of inputs. The three different types of clustering architectures will prove their importance in the paradigm of wireless sensor networks. The architectural strategy of ART1 and Fuzzy ART makes them easily suitable for VLSI implementation.

REFERENCE

1. Sudhir G. Akojwar and Rajendra. M. Patrikar, “Neural Networks based Real Time Classifier for Wireless Sensor Networks and Framework for VLSI Implementation”, 1-4244-0755-9/07, pp. 1381-1386 , 4-7 June, 2007 IEEE
2. A Wendi B. Heinzelman, Ananth P. Chandrakasan, and Hari Balakrishnan, An Application -Specific protocol Architecture for Wireless Microsensor Networks, IEEE Transaction on Wireless Communications, Vol.1 No. 4 October 2002
3. Network For Supervised Learning, Recognition, and Prediction, IEEE Communications Magazine, Sep 1992
4. Teresa Serrano-Gotarrendona and Bemabe Linares-barranco, An ART1 Microchip and its use in Multi-ART1 systems, IEEE Transaction on Neural Networks, VOL 8, No. 5, Sep 1997.
5. G. Carpenter, S. Grossberg, D. Rosen, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," Neural Networks, Vol. 4, pp. 759-771, 1991.
6. C. C. Aggrawal, J. Han, and P. S. Yu. “On Demand Classification of Data Streams”, In KDD (2004), pp.503-508
7. M. W. Kadous and C. Sammut. “Classification of multivariate time series and structured data using constructive induction. Machine Learning”, Journal (2005), pp.176-216
8. Sudhir G. Akojwar, R.M.Patrikar, Classification techniques for sensor data and clustering architecture for wireless sensor networks, IAENG International conference on communication systems and applications (ICCSA'07) ,Hong Kong, 21-23 march, 2007.
9. J. Lin, E. Keogh, S. Lonardi, and B. Chiu. “A Symbolic Representation of Time Series with Implications for Streaming Algorithms”, In DMKD (2003), pp.2-11
10. J.A.Freeman, David M. Skapura, “Neural networks Algorithms, applications, and programming techniques”, Pearson Education Asia, 2001.
11. Gail A. Carpenter and Stephen Grossberg, A Self-Organising Neural Network For Supervised Learning, Recognition, and Prediction, IEEE Communications Magazine, Sep 1992
12. Sudhir G. Akojwar, R.M.Patrikar, Neural networks for Wireless Sensor Networks, IMAPS India National Conference on Microelectronics and VLSI (IINC 2005), IIT Bombay, INDIA, 19-21 Dec 2005.